

Package ‘gprege’

October 16, 2018

Version 1.24.0

Date 2016-10-07

Title Gaussian Process Ranking and Estimation of Gene Expression
time-series

Author Alfredo Kalaitzis <alkalait@gmail.com>

Maintainer Alfredo Kalaitzis <alkalait@gmail.com>

Depends R (>= 2.10), gptk

Suggests spam

Description The gprege package implements the methodology described in Kalaitzis & Lawrence (2011) "A simple approach to ranking differentially expressed gene expression time-courses through Gaussian process regression". The software fits two GPs with the an RBF (+ noise diagonal) kernel on each profile. One GP kernel is initialised with a short lengthscale hyperparameter, signal variance as the observed variance and a zero noise variance. It is optimised via scaled conjugate gradients (netlab). A second GP has fixed hyperparameters: zero inverse-width, zero signal variance and noise variance as the observed variance. The log-ratio of marginal likelihoods of the two hypotheses acts as a score of differential expression for the profile. Comparison via ROC curves is performed against BATS (Angelini et.al, 2007). A detailed discussion of the ranking approach and dataset used can be found in the paper (<http://www.biomedcentral.com/1471-2105/12/180>).

License AGPL-3

biocViews Microarray, Preprocessing, Bioinformatics,
DifferentialExpression, TimeCourse

URL

BugReports alkalait@gmail.com

git_url <https://git.bioconductor.org/packages/gprege>

git_branch RELEASE_3_7

git_last_commit 1495611

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

R topics documented:

gprege-package	2
compareROC	3
DellaGattaData	3
demTp63Gp1	4
DGdat_p63	5
exhaustivePlot	5
gprege	6
rocStats	8

Index	9
--------------	----------

gprege-package	<i>gprege - Gaussian Process Ranking and Estimation of Gene Expression.</i>
----------------	---

Description

This package implements the method of Kalaitzis and Lawrence (2011) for Gaussian process modelling gene expression time-series data. The method can be used to filter quiet genes and quantify differential expression in time-series expression ratios.

Details

```

Package: gprege
Type: Package
Version: 0.99.0
Date: 2011-07-08
License: A-GPL Version 3

```

For details of using the package please refer to the Vignette.

Author(s)

Alfredo Kalaitzis
 Maintainer: Alfredo Kalaitzis <alkalait@gmail.com>

References

A.~A.~Kalaitzis and N.~D.~Lawrence. A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression *BMC Bioinformatics* 2011, 12:180. DOI:10.1186/1471-2105-12-180.

See Also

[demGpCov2D](#), [demGpSample](#), [demInterpolation](#), [demOptimiseGp](#), [demRegression](#)

Examples

```
## see demTp63Gp1.R
```

compareROC	<i>Make ROC plots.</i>
------------	------------------------

Description

This rocStats wrapper superimposes ROC curves on a plot to analyse the output performance of a method-A, and optionally compare it with that of a method-B, based on some ground truth labels.

Usage

```
compareROC(output, groundTruthLabels, compareToRanking)
```

Arguments

output	(vector) The output of ranking scores returned by method-A for each data-point.
groundTruthLabels	(vector) A binary vector that contains the ground truth (e.g. which genes are members of the top-100 ground truth list).
compareToRanking	A matrix where each column is the output vector of ranking scores returned by another competing method.

Value

area	A scalar. The area under the ROC curve of method-A.
------	---

See Also

[rocStats](#)

Examples

```
data(FragmentDellaGattaData) ## Load demo data.
compareROC(output= rnorm(length(DGatta_labels_byTSNI))>0, groundTruthLabels=DGatta_labels_byTSNI)
```

DellaGattaData	<i>Fragment dataset of 13 time-point mouse microarray time series of gene expression ratios and a ranking list of TP63 targets suggested by TSNI.</i>
----------------	---

Description

exprs_tp63_RMA 100 gene reporters of 13 time-points mouse Affymetrix microarray gene expression coming from a study on primary mouse keratinocytes with an induced activation of the TRP63 transcription factor (GEO-accession number:GSE10562, see Source section), where a reverse-engineering algorithm was developed (TSNI: time-series network identification) to infer the direct targets of TRP63 (Della Gatta et.al. 2008). The data has been processed using [rma](#) (affy) and the profiles are centred (zero-mean) across the timepoints.

DGatta_labels_byTSNI, DGatta_labels_byTSNItop100 a ranking list suggested based by TSNI is provided. The inferred direct targets were biologically confirmed by correlation with ChIP-Seq binding regions; therefore the list is used as a noisy ground truth. See Source section.

genesymbols Names of the genes that the transcript_IDs (in `exprs_tp63_RMA`) correspond to.

gpregeOutput Its field 'rankingScores' contains log-marginal likelihood ratios, used as ranking scores, for each gene reporter in `exprs_tp63_RMA`. This is the output from a run of `gprege` on the full DellaGatta dataset (see `demTp63Gp1.R`) and stored here for convenience.

Usage

```
data(FragmentDellaGattaData)
```

Source

GEO: <http://www.ncbi.nlm.nih.gov/geo/>, TSNI ranking: genome.cshlp.org/content/suppl/2008/05/05/gr.073601.107.DC1/DellaGatta_SupTable1.xls

References

Della Gatta G, et al. Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Research* 2008, 18(6):939.

demTp63Gp1

gprege on TP63 expression time-series.

Description

Demo script of Gaussian Process Regression and Estimation of Gene Expression on TP63 time-series data (see `gprege.m`). See Kalaitzis & Lawrence (2011) for a detailed discussion of the ranking algorithm and dataset used.

Usage

```
demTp63Gp1(fulldataset=FALSE)
```

Arguments

`fulldataset` (Logical) TRUE downloads and uses the full dataset.

See Also

[gprege](#)

Examples

```
demTp63Gp1(fulldataset=FALSE)
```

DGdat_p63	<i>BATS rankings (Angelini, 2007) Case 1: Delta error prior Case 2: Inverse Gamma error prior Case 3: Double Exponential error prior</i>
-----------	--

Description

DGdat_p63_case1_GL Case 1: Delta error prior
DGdat_p63_case2_GL Case 2: Inverse Gamma error prior
DGdat_p63_case3_GL Case 3: Double Exponential error prior

Usage

```
data(DGdat_p63)
```

Source

GEO: <http://www.ncbi.nlm.nih.gov/geo/>, TSNI ranking: genome.cshlp.org/content/suppl/2008/05/05/gr.073601.107.DC1/DellaGatta_SupTable1.xls

References

Della Gatta G, et al. Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Research* 2008, 18(6):939.

exhaustivePlot	<i>Plot of the LML function by exhaustive search.</i>
----------------	---

Description

Exhaustively searches the hyperparameter space by a grid, whose resolution is passed as an argument, and plots the LML function for every point in the space.

Usage

```
exhaustivePlot(y, x, xstar, options, maxwidth, res, nlevels)
```

Arguments

y	the target (output) data.
x	the input data matrix.
xstar	the points to predict function values.
options	options structure as defined by gpOptions.m.
maxwidth	maximum lengthscale to search for.
res	The search resolution. Number of points to plot for in the search range.
nlevels	Number of contour levels.

Value

area Area under the ROC curve of method-A.

See Also

[rocStats](#)

Examples

```
noiseLevel <- 0.2
noiseVar <- noiseLevel^2
options <- gpOptions()
options$kern$comp <- list('rbf','white')
## Create data set
l <- 9; x <- matrix(seq(0,240,by=20), ncol=1)
trueKern <- kernCreate(x, 'rbf')
trueKern$inverseWidth <- 1/(20^2) ## Characteristic inverse-width.
K <- kernCompute(trueKern, x) + diag(dim(x)[1])*noiseVar
## Sample some true function values.
y <- gaussSamp(Sigma=K, numSamps=1)
xTest <- as.matrix(seq(0, 240, length=200))
graphics.off(); dev.new(); plot.new(); dev.new(); plot.new()
exhaustivePlot(y, x, xTest, options=options, maxwidth=100, res=50, nlevels=75)
```

gprege

Gaussian process ranking and estimation of gene expression time-series

Description

Fits two GPs with the an RBF (+ noise diagonal) kernel on each profile. One GP kernel is initialised with a short lengthscale hyperparameter, signal variance as the observed variance and a zero noise variance. It is optimised via scaled conjugate gradients (netlab). The other GP has fixed hyperparameters with a zero inverse-width, zero signal variance and noise variance as the observed variance. The log-ratio of marginal likelihoods of the two hypotheses acts as a score of differential expression for the profile. Comparison via ROC curves is performed against BATS (Angelini et.al, 2007). See Kalaitzis & Lawrence (2011) for a detailed discussion of the ranking algorithm and dataset used.

Usage

```
gprege(data, inputs, gpregeOptions)
```

Arguments

data The matrix of gene expression profiles; one profile per row.

inputs Inputs (timepoints) to the GP.

gpregeOptions Options list for gprege with fields

explore Logical. TRUE operates in a user interactive mode. Used for examining individual gene expression profiles.

labels A binary vector. TRUE specifies whether the corresponding profile comes from a differentially expressed gene (usually from a ground truth).

- indexRange** A numeric vector. Range of indices of profiles on which the function should operate. Useful for selective exploration of specific profiles, e.g. only genes marked as differentially expressed in a ground truth list.
- interpolatedT** A numeric vector. New timepoints to interpolate for each profile, based on the estimated function values.
- iters** A scalar. The number of iterations for scaled-conjugate gradients (SCG) optimisation.
- display** Logical. Display gradient and LML information on each SCG iteration.
- inithypers** The matrix of hyperparameter configurations as its rows. Each row has the following format: [inverse-lengthscale percent-signal-variance percent-noise-variance] The first row corresponds to a (practically constant) function with a very large lengthscale. Such a function will account for 0 percent of the observed variance in the expression profile (hence 0 for signal) and explain it as noise (hence 1 for noise). Subsequent rows (initialisations for SCG optimisation) correspond to functions of various lengthscales that explain all the observed variance as signal. A reasonable lengthscale would be roughly in line with the time-point sampling intervals.
- exhaustPlotRes** A scalar. The search resolution. Used for interactive mode (explore == 1).
- exhaustPlotLevels** A scalar. Number of contour levels in the exhaustive plot. Used for interactive mode (explore == 1).
- exhaustPlotMaxWidth** A scalar. the maximum lengthscale to search for. Used for interactive mode (explore == 1).

Value

- `gpregeOutput` Output list with fields:
- signalvar** A numeric vector of the vertical lengthscales of the optimised RBF kernel; one for each profile.
- noisevar** A numeric vector. Similar to `signalvar`, but for the noise hyperparameter.
- width** A numeric vector. Similar to `signalvar` and `noisevar`, but for the horizontal lengthscales of the RBF.
- LMLs** A numeric vector of log-marginal likelihoods of the GP; one for each profile.
- interpolatedData** A matrix of the extended dataset with interpolated values as the augmenting columns.
- rankingScores** A numeric vector of the ranking scores, based on the log-ratio of marginal likelihoods.

See Also

[gpOptions](#), [gpCreate](#), [gpExpandParam](#), [gpOptimise](#), [gpExtractParam](#), [gpLogLikelihood](#), [gpPosterior](#)

Examples

```
## see demTp63Gp1.R
data(FragmentDellaGattaData) ## Load demo data.
## Setup other gprege options.
gpregeOptions = list(indexRange=(1:2), explore=TRUE, exhaustPlotRes=30, exhaustPlotLevels=10,
```

```

    exhaustPlotMaxWidth=100, iters=100, labels=DGatta_labels_byTSNI, display=FALSE)
## Matrix of different hyperparameter configurations as rows:
## [inverse-lengthscale  percent-signal-variance  percent-noise-variance].
gpregeOptions$inithypers <- matrix( c(
  1/1000,1e-3,0.999
  ,1/20,0.999,1e-3
  ), ncol=3, byrow=TRUE)
gpregeOutput <- gprege(data=exprs_tp63_RMA, inputs=matrix(seq(0,240,by=20), ncol=1), gpregeOptions=gpregeO

```

rocStats

Make ROC curve data.

Description

Computes the points on an ROC curve by varying a threshold on the sorted outputs of the method in question.

Usage

```
rocStats(outputs, groundTruthLabels, decreasing = TRUE)
```

Arguments

outputs	A numeric vector with the outputs of the evaluated method (e.g. likelihoods from gprege).
groundTruthLabels	A binary vector than contains the ground truth (e.g. which genes belong in the top-100 ground truth list).
decreasing	Logical. TRUE sorts outputs by decreasing order.

Value

stats	A list of numeric variables with the necessary statistics to compute an ROC curve, a precision-recall curve, etc.
-------	---

Examples

```

## see compareROC.R
data(FragmentDellaGattaData) ## Load demo data.
rocStats(gpregeOutput$rankingScores, DGatta_labels_byTSNItop100, decreasing=TRUE)

```


Index

*Topic **datasets**

DellaGattaData, [3](#)

DGdat_p63, [5](#)

*Topic **package**

gprege-package, [2](#)

compareROC, [3](#)

DellaGattaData, [3](#)

demGpCov2D, [2](#)

demGpSample, [2](#)

demInterpolation, [2](#)

demOptimiseGp, [2](#)

demRegression, [2](#)

demTp63Gp1, [4](#)

DGatta_labels_byTSNI (DellaGattaData), [3](#)

DGatta_labels_byTSNItop100
(DellaGattaData), [3](#)

DGdat_p63, [5](#)

DGdat_p63_case1_GL (DGdat_p63), [5](#)

DGdat_p63_case2_GL (DGdat_p63), [5](#)

DGdat_p63_case3_GL (DGdat_p63), [5](#)

exhaustivePlot, [5](#)

exprs_tp63_RMA (DellaGattaData), [3](#)

FragmentDellaGattaData

(DellaGattaData), [3](#)

genesymbols (DellaGattaData), [3](#)

gpCreate, [7](#)

gpExpandParam, [7](#)

gpExtractParam, [7](#)

gpLogLikelihood, [7](#)

gpOptimise, [7](#)

gpOptions, [7](#)

gpPosteriorMeanVar, [7](#)

gprege, [4](#), [6](#)

gprege-package, [2](#)

gpregeOutput (DellaGattaData), [3](#)

rma, [3](#)

rocStats, [3](#), [6](#), [8](#)